



jamk.fi

Selaimella pelattavan oppimispelin jatkokehitys

Jussi Pietiläinen

Opinnäytetyö

Maaliskuu 2016

Tekniikan ja liikenteen ala

Insinööri (AMK), mediatekniikan koulutusohjelma

Jyväskylän ammattikorkeakoulu

JAMK University of Applied Sciences

Tekijä(t) Pietiläinen, Jussi	Julkaisun laji Opinnäytetyö, AMK	Päivämäärä Maaliskuu 2016
	Sivumäärä 47	Julkaisun kieli Suomi
		Verkojulkaisulupa myönnetty: x
Työn nimi Selaimella pelattavan oppimispelin jatkokehitys		
Tutkinto-ohjelma Mediatekniikan koulutusohjelma		
Työn ohjaaja(t) Kari Niemi		
Toimeksiantaja(t) Onerva Mäen koulu		
<p>Tiivistelmä</p> <p>Työn toimeksiantajana toimi Onerva Mäen koulu. Tehtävänä oli jatkaa edellisenä lukuvuonna ”Projekti”-opintojaksolla pelattavaan kuntoon saadun opetuspelin kehittämistä. Peliä oli tarkoitus parantaa korjaamalla löydetty virheet ja optimoimalla sovellusta. Lisäksi peliin haluttiin lisätä muutama ominaisuus, joista suurimmat olivat palkintokauppa ja sisällönhallintatyökalu. Palkintokaupasta pelaaja voisi ostaa ansaitsemillaan pisteillä erilaisia palkintoja ja sisällönhallintatyökalulla käyttäjä pystyisi helposti muokkaamaan pelin sisältöä ja jopa tekemään saman pelirungon päälle kokonaan uuden pelin. Päättävöitteena oli saada peli julkaisukelpoiseen kuntoon.</p> <p>Käytännön tehtävien lisäksi haluttiin myös yleisemmän tason tutkimus verkkosovellusoptimoinnista.</p> <p>Optimointiosuutta tutkittiin ja opiskeltiin ensin eri lähteistä. Tämän jälkeen opittuja keinoja toteutettiin käytännössä pelin kehittämisessä. Palkintokaupan kehittämisessä hyödynnettiin tuttuja tekniikoita sovellukseen jo aiemmin luoduista osioista. Sisällönhallintatyökalu tehtiin PHP:n avulla. Työn aikana oltiin yhteydessä toimeksiantajan edustajiin ja kehitystä vietiin eteenpäin heiltä saadun palautteen perusteella. Yhteysvälineinä käytettiin pääosin Yammeria ja sähköpostia, mutta myös tapaamisia järjestettiin.</p> <p>Työn tuloksena saatiin yleispätevä raportti verkkosovelluksen optimoinnista sekä julkaisukelpoinen jatkokehitetty oppimispeli. Asetetut vähimmäistavoitteet siis saavutettiin. Teoriaosuus optimoinnista jäi ajanpuutteen vuoksi jonkin verran suppeammaksi kuin oli alunperin tarkoitus.</p>		
Avainsanat (asiasanat) verkkosovellusoptimointi		
Muut tiedot		

Author(s) Pietiläinen, Jussi	Type of publication Bachelor's thesis	Date March 2016
	Number of pages 47	Language of publication: Finnish
		Permission for web publication: x
Title of publication Further development for a web browser learning game		
Degree programme Media engineering		
Supervisor(s) Kari Niemi		
Assigned by Onerva Mäen koulu		
<p>Description</p> <p>The subject for this thesis was given by Onerva Mäki's school. The purpose was to fix and continue developing the learning game that had been created last semester. All the bugs were needed to be fixed and the application had to be optimized. Besides that also some new features were wanted. The game should include a prize shop and some kind of administration tool that could be used for editing the game contents and even to create new games. The main goal was to get a publishable version of the game so Onerva could add it to their home pages.</p> <p>Also a common study of web application optimization was wanted from the thesis.</p> <p>Optimization techniques were first studied from different sources and then the techniques were tested in practice and used for optimizing the game. For developing the prize shop the techniques from the former parts of the game were used. The administration tool was made with PHP.</p> <p>During the work some feedback was received from the client. The main communication channels with the client were Yammer and e-mail but also some live meetings were hold.</p> <p>As a result of the thesis a report of web application techniques was made. Also the game was fixed and optimized to be good to publish. The new features were added as planned. The thesis wasn't as extensive as originally planned but the main goals for the work were reached.</p>		
Keywords (subjects)		
web optimization		
Miscellaneous		

Sisältö

Sanasto	4
1 Työn lähtökohdat	6
1.1 Toimeksiantaja ja taustaa	6
1.2 Tehtävä ja tavoitteet	9
2 Web-sovellusten optimointi ja suorituskykymittaus	9
2.1 Yleistä	9
2.2 Palvelinpuolen optimointi	10
2.2.1 Tiedostojen jakaminen eri palvelimille	10
2.2.2 Tiedostojen GZIP-pakkaaminen	11
2.3 HTTP-pyyntömäärien minimointi	12
2.4 HTML-, CSS- ja Javascript-tiedostojen minifiointi	13
2.5 Kuvien optimointi	13
2.5.1 Kuvien korvaaminen	13
2.5.2 Oikean kuvaformaatin valinta	13
2.5.3 Oikean kokoisen kuvan käyttäminen	14
2.5.4 Kuvaoptimointisovellusten hyödyntäminen	15
2.5.5 Vektorigrafiikan käyttäminen	15
2.6 Selaimen välimuistin hyödyntäminen	15
2.7 Apuvälineet	16
2.7.1 Google PageSpeed Insights	16
2.7.2 Yahoo!n YSlow	17
2.7.3 Verkkoselainten omat kehitystyökalut	19
3 Riimitaituri Orre Oravan suorituskykyoptimointi ja bugikorjaukset	20
3.1 Ääniin liittyvien ongelmien korjaaminen	20
3.2 Pätkivän palkintovideon korjaaminen	20
3.3 iPad-ongelmien korjaaminen	21

3.4	Javascript-tiedostojen minimointi ja yhdistäminen	21
3.5	Kuvien optimointi	23
3.6	Palvelinpuolen optimointi	24
3.7	Testitulokset	25
3.8	Raskaiden 3D-mallien ongelma	25
4	Sovelluksen jatkokehitys	27
4.1	Pienet lisätoiminnot	27
4.1.1	Alkuvideon ohitusmahdollisuus	27
4.1.2	Tehtäväsiirtymän ohitusmahdollisuus	27
4.1.3	Mobiililaitteilla pystyasennossa pelaamisen estäminen.....	28
4.2	Palkintokauppa	28
4.3	Työkalu uuden sisällön tuottamiseen.....	32
4.3.1	Perusrakenne ja käytetyt tekniikat	33
4.3.2	Tekstien muokkaaminen	33
4.3.3	Kuvien muokkaaminen.....	34
4.3.4	Tehtävärastien valinta.....	36
4.3.5	Tehtävä-äänien lataaminen	37
4.3.6	Muiden äänien lataaminen	39
4.3.7	Videoiden lataaminen	39
4.3.8	Palkintojen hintojen muuttaminen	40
4.3.9	Sisäänkirjautuminen.....	41
5	Tulokset ja loppupohdinta	42
	Lähteet.....	44

Kuviot

Kuvio 1. Pelin alkuvalikko	7
Kuvio 2. Pelin yleisnäkymä	8
Kuvio 3. Tehtävärastinäkymä	8

Kuvio 4. Tiedoston lähettäminen palvelimelta selaimelle GZIP-pakattuna	11
Kuvio 5. Kuvakaappaus PageSpeed Insights –analyysin tulossivusta	17
Kuvio 6. Kuvakaappaus YSlow-analyysin tulossivusta.....	18
Kuvio 7. Kuvakaappaus Pingdom Tools –analyysin tulossivusta.....	19
Kuvio 8. GZIP-pakkaustestin tulossivu.....	24
Kuvio 9. Kaupan pohjakuva ilman palkintoja	29
Kuvio 10. Kauppa kaikkien palkintojen kanssa.....	30
Kuvio 11. Palkintokaupan yksittäinen kuvakerros: Orre ilman ”etukättä”	31
Kuvio 12. Pelin kaupanäkymä	32
Kuvio 13. ”Tekstit”-välilehti sisällönhallintatyökalussa.....	34
Kuvio 14. ”Kuvat”-välilehti sisällönhallintatyökalussa	35
Kuvio 15. ”Tehtävät”-välilehti sisällönhallintatyökalussa	36
Kuvio 16. Tehtävärastien tallentaminen ja lataaminen	37
Kuvio 17. ”Tehtävä-äänet”-välilehti sisällönhallintatyökalussa	38
Kuvio 18. ”Muut äänet”-välilehti sisällönhallintatyökalussa	39
Kuvio 19. ”Videot”-välilehti sisällönhallintatyökalussa.....	40
Kuvio 20. ”Palkinnot”-välilehti sisällönhallintatyökalussa	41
Kuvio 21. Sisällönhallintatyökalun sisäänkirjautumisenäkymä.....	42

Taulukot

Taulukko 1. Kuvaformaattien ominaisuudet.....	14
Taulukko 2. JS-tiedostojen viemän muistin määrä ennen ja jälkeen minimoinnin	22
Taulukko 3. Kuvien viemän muistin määrä ennen ja jälkeen optimoinnin.....	23

SANASTO

CPU

Central Processing Unit (suoritin, prosessori). Tietokoneen osa, joka suorittaa tietokoneohjelman sisältämiä konekielisiä käskyjä.

CSS

Cascading Style Sheets. Antaa tyyliohjeet WWW-dokumenttien sisällön näyttämiseksi.

Div

HTML-elementti, jonka avulla voi ryhmitellä sisältöjä ja antaa niille attribuutteja.

Front-end

Asiakkaalle verkkosovelluksesta näkyvä osa, jonka parissa asiakas käytännössä toimii. Osa-alueeseen kuuluvat HTML, CSS ja Javascript.

GIF

Graphic Interchange Format. Häviötön bittikarttagrafiikan tallennusformaatti.

GZIP

GNU zip. Tiedostojen pakkausohjelma, jolla tiedostot saadaan tiivistettyä pienempään tilaan.

Htaccess

Hypertext Access. Apachen hakemistokohtainen asetustiedosto, jolla voidaan ylikirjoittaa Apachen pääasetustiedoston sääntöjä

HTML

Hypertext Markup Language. Merkkauskieli, jolla mm. verkkosivut on pääosin koodattu.

HTTP

Hypertext Transfer Protocol. protokolla, jota selaimet ja WWW-palvelimet käyttävät tiedonsiirtoon.

Javascript

Pääasiassa verkkoympäristössä käytettävä dynaaminen komentosarjakieli.

Javascript-kirjastot

Javascript-ohjelmointia helpottavat valmiit kirjastot, jotka sisältävät valmiita funktioita ja objekteja.

JPEG

Joint Photographic Experts Group. Häviöllistä pakkausta käyttävä bittikarttagrafiikan tallennusformaatti.

JQuery

Kaikille selaimille tarkoitettu ilmainen, avoimen lähdekoodin JavaScript-kirjasto.

JSON

JavaScript Object Notation. Yksinkertainen avoimen standardin tiedostomuoto tiedonvälitykseen.

Kehys (frame)

Keino, jonka avulla verkkosivu voidaan jakaa osiin, joissa voidaan näyttää eri HTML-sivujen sisältöjä.

Palvelin

Tietokoneessa suoritettava palvelinohjelmisto ja ohjelmistoa suorittava tietokone, jonka tehtävänä on tarjota palveluita muille ohjelmille.

PHP

PHP: Hypertext Preprocessor). Ohjelmointikieli, jota käytetään erityisesti Web-palvelinympäristöissä dynaamisten web-sivujen luonnissa.

PNG

Portable Network Graphics. Häviötön bittikarttagrafiikan tallennusformaatti.

SVG

Scalable Vector Graphics. Kaksiulotteisten vektorikuvien kuvauskieli.

Three.js

Javascript-kirjasto/API, jonka avulla voidaan luoda ja näyttää animoitua 3D-grafiikkaa verkkoselaimella.

1 TYÖN LÄHTÖKOHDAT

1.1 Toimeksiantaja ja taustaa

Projektin toimeksiantaja oli Oppimis- ja ohjauskeskus Valteriin kuuluva Onerva. Onerva tarjoaa palveluja oppimisen ja koulunkäynnin tuen tarpeisiin. Erityistä asiantuntemusta Onervalla on tarjota näkemiseen, kuulemiseen, kieleen ja vuorovaikutukseen liittyvistä tuen tarpeista. Onervan toimitilat sijaitsevat Jyväskylässä. (Valteri-koulu Onerva n.d.)

Alkuperäinen projekti alkoi jo syksyllä 2014 Jyväskylän ammattikorkeakoulun ”Projekti”-opintojaksolla. Tuolloin viisihenkiselle mediatekniikan kolmannen vuoden opiskelijoista koostuvalle projektiryhmälle annettiin aiheeksi luoda verkkoselaimella pelattava riimittelypeli. Pelin oli tarkoitus toimia niin pöytätietokoneilla kuin myös kohtuullisen nykyaikaisilla tablet-laitteilla ilman erillisiä lisäosa-asennuksia. Lisäksi pelin tuli jollain tavalla esitellä vuoden 2015 lopulla valmistuvaa Onervan uutta toimitilaa. Muutoin projektiryhmälle annettiin varsin vapaat kädet sovelluksen toteuttamiseen.

Projektissa hyödynnettiin three.js-javascript-kirjastoa, jonka avulla peli saatiin toimimaan 3D-maailmassa. Pelissä pääsee liikkumaan Onervan uuden koulurakennuksen tiloissa samalla, kun ratkoo aiheisiin liittyviä riimittelystehtäviä. Oikeista vastauksista pelaajaa palkitaan omenoilla ja pelikierroksen lopuksi pelaajalle näytetään loppuvideo sen mukaan, miten hyvin tämä on pärjännyt. Pelin nimeksi tuli lopulta Riimitaituri Orre Orava.

Peli saatiin jo opintojakson aikana pelattavaan kuntoon, mutta siihen jäi vielä muutamia virheitä ja sovellus oli myös hieman turhan raskas. Lisäksi toimeksiantajalla oli vielä toiveissa saada peliin muutama lisäominaisuus, joten näistä lähtökohdista jatkettiin opinnäytetyön merkeissä.

Pelin alussa käyttäjälle näytetään alkuvalikko (ks. Kuvio 1.). Jos samalla selaimella on jo aiemmin pelattu ja peli on jäänyt kesken, tarjotaan käyttäjälle mahdollisuus jatkaa

aiempaa peliä. ”Uusi peli” –painikkeesta voi aloittaa kokonaan uuden pelin milloin vain ja ”Info” –painikkeesta avautuvat pelin tiedot.



Kuvio 1. Pelin alkuvalikko

Pelin yleisnäkymässä (ks. Kuvio 2.) käyttäjä näkee valittavissa olevat ja jo suoritettut tehtävärastit. Käymättömät rastit ovat kokonaisia omenoita, joista klikkaamalla pääsee tehtävään. Kun tehtävä on suoritettu, muuttuu kyseinen omena syödyksi. Vasemmassa reunassa olevasta kerrosvalikosta käyttäjä voi valita aktiivisena näytettävän kerroksen. Alareunan valikossa näkyvät infopainike, kerättyjen pisteiden määrä ja kauppapainike.



Kuvio 2. Pelin yleisnäkymä

Tehtäväraстille siirryttäessä avautuu käyttäjälle tehtäväraстinäkymä (ks. Kuvio 3.). Alussa kuuluu tehtävä, jonka voi halutessaan kuunnella uudestaan. Vastausvaihtoehdot voi kuunnella klikkaamalla alareunan omenia. Vastaaminen tapahtuu raahaamalla oikea omena "?"-omenan päälle.



Kuvio 3. Tehtäväraстinäkymä

1.2 Tehtävä ja tavoitteet

Opinnäytetyön tehtävät ja tavoitteet jakautuivat kahteen isompaan kokonaisuuteen. Ensimmäisessä vaiheessa oli tarkoitus korjata jo olemassa olevan sovelluksen virheet sekä erityisesti parantaa sen suorituskykyä optimoinnin avulla. Työssä perehdyttiin Web-sovellusten optimointiin ensin yleisellä tasolla ja tämän jälkeen hyödynnettiin saatuja oppeja käytännössä optimoimalla Riimitaituri Orre Orava –peliä.

Opinnäytetyön toinen isompi asiakokonaisuus oli saman pelin jatkokehitys. Peliin haluttiin luoda ”palkintokauppa”, josta pelin päähenkilö Orrelle pystyy ostamaan erilaisia palkintoja ansaittujen omenapisteiden avulla. Tämän osion tekemistä oli suunniteltu jo Projekti-opintojaksolla, mutta sen toteuttamiseen ei lopulta jäänyt aikaa. Lähes kaikki grafiikat tähän oli kuitenkin luotu jo etukäteen.

Toisena jatkokehityskohteena oli luoda jonkinlainen hallinnointipaneeli, jonka kautta pelin sisällöt olisivat helposti muokattavissa. Sen avulla pystyisi siis muokkaamaan nykyistä peliä tai jopa luomaan samalle pohjalle kokonaan uuden pelin. Tätä ei ollut edellisellä opintojaksolla suunniteltu lainkaan etukäteen.

Yhtenä kantavana ajatuksena oli myös pitää peli/pelin runko mahdollisimman helppona käyttöönotettavana. Eli mielellään pelin saisi helpommillaan käyttöön vain laittamalla tiedostot palvelimelle eikä mitään muita asennuksia tai säätöjä tarvitsisi palvelinpuolella tehdä.

2 WEB-SOVELLUSTEN OPTIMOINTI JA SUORITUSKYKYMITTAUS

2.1 Yleistä

Web-sovelluksen suorituskyky on tärkeä osa käyttäjäkokemusta. Kaikki ylimääräinen odottelu ja latailu heikentää sovelluksen käyttömukavuutta. Useat tutkimukset osoittavat, että jo pienetkin erot sivujen latausajoissa näkyvät kävijämäärissä.

Esimerkiksi Amazon raportoi jo vuonna 2007, että jokainen sekunnin kymmenyksen lisäys latausaikoihin vähensi myyntiä yhdellä prosentilla. Facebook puolestaan on ilmoittanut, että puolen sekunnin viive sivuilla aiheuttaa 3 % ja sekunnin viive 6 % vähemmän liikennettä. Yleisiä huomiota ovat myös, että 47 % käyttäjistä olettaa sivun latautuvan alle kahdessa sekunnissa ja jopa 40 % poistuu sivulta, jos latautuminen kestää yli kolme sekuntia. (Guasch 2012.)

Web-sovelluksen optimointia voidaan tehdä niin palvelin- kuin myös asiakaspään puolella.

2.2 Palvelinpuolen optimointi

2.2.1 Tiedostojen jakaminen eri palvelimille

Joissain tapauksissa sivujen latausaikoja on mahdollista lyhentää asettamalla sivujen staattisia komponentteja kuten kuvia sekä CSS- ja JS-tiedostoja eri palvelimille. Tämä johtuu siitä, että useimmat selaimet voivat ladata vain rajoitettua määrää tiedostoja samanaikaisesti yksittäiseltä palvelimelta. Siispä jos tiedostot on jaettu useammalle palvelimelle, voidaan sitä suurempaa määrää ladata yhtä aikaa. (Rumpel 2013.)

Huono puoli tässä on, että IP-osoitteiden etsimiseen menee oma aikansa ja tämä hieman kuluttaa saavutettua etua. Uusimmilla selaimilla voi hakea IP-osoitteita jo etukäteen tekemällä tarvittavan "dns-prefetch" –viittauksen lähdekoodissa. Tällöin etsimiseen ei mene aikaa enää siinä vaiheessa, kun tiedostoa aletaan lataamaan. Tällaisen viittauksen pystyy tekemään seuraavan esimerkin mukaisesti:

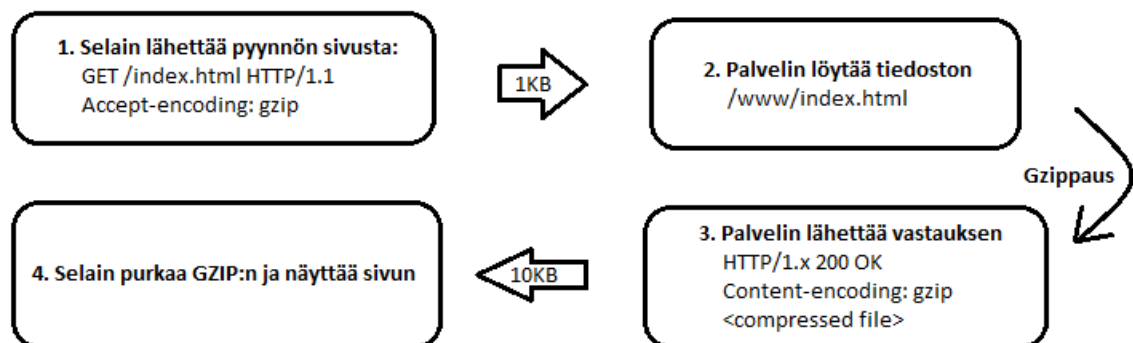
```
<link href="//another-domain.com" rel="dns-prefetch" />
```

(Rumpel 2013.)

2.2.2 Tiedostojen GZIP-pakkaaminen

Tiedostojen kokoa on mahdollista pienentää huomattavasti lataamisen ajaksi pakkaamalla ne GZIP-muotoon. Tätä toimintoa varten pitää tiedostoille asettaa pakkaamiseen liittyvät määrytykset palvelinpuolella. (Sexton, 2015.)

Käytännössä pakkaaminen toimii kuvion 4. osoittamalla tavalla. Käyttäjän selain lähettää palvelimelle pyynnön, jossa se ilmoittaa hyväksyvänsä myös GZIP-pakatut tiedostot. Palvelin vastaanottaa pyynnön ja huomaa maininnan pakkausmahdollisuudesta. Palvelinpuolella tiedosto ensin pakataan ja sitten se lähetetään pakattuna selaimelle. Selain vastaanottaa pakatun tiedoston, purkaa sen ja lopuksi näyttää puretun sisällön käyttäjälle. Mikäli selaimen lähettämässä pyynnössä ei ole mainintaa GZIP-mahdollisuudesta ja/tai palvelimella ei ole GZIP-määrytykset käytössä, tällöin tiedostoa ei pakata. (Sexton, 2015.)



Kuvio 4. Tiedoston lähettäminen palvelimelta selaimelle GZIP-pakattuna

Pakkaamisen aktivointi määritellään eri palvelimilla omissa konfiguraatio-tiedostoissa. Esimerkiksi Apache-palvelimella määrittelyn voi tehdä htaccess-tiedostoon lisäämällä seuraavat rivit:

```
<ifModule mod_gzip.c>
mod_gzip_on Yes
mod_gzip_dechunk Yes
mod_gzip_item_include file .(html?|txt|css|js|php|pl)$
mod_gzip_item_include handler ^cgi-script$
mod_gzip_item_include mime ^text/.
mod_gzip_item_include mime ^application/x-javascript.*
```

```
mod gzip item exclude mime ^image/.*
mod_gzip_item_exclude rspheader ^Content-Encoding:.*gzip.*
</ifModule>
```

Tai jos pakkaaminen ei edellä mainituilla määrittäyksillä toimi, voidaan se tehdä vaihtoehtoisesti näillä riveillä:

```
AddOutputFilterByType DEFLATE text/plain
AddOutputFilterByType DEFLATE text/html
AddOutputFilterByType DEFLATE text/xml
AddOutputFilterByType DEFLATE text/css
AddOutputFilterByType DEFLATE application/xml
AddOutputFilterByType DEFLATE application/xhtml+xml
AddOutputFilterByType DEFLATE application/rss+xml
AddOutputFilterByType DEFLATE application/javascript
AddOutputFilterByType DEFLATE application/x-javascript
```

GZIP-pakkaamisen toimivuuden voi testata nopeasti jollakin WEB-sovelluksella, esimerkiksi tällä sivulla: <https://varvy.com/tools/gzip/>. (Sexton, 2015.)

2.3 HTTP-pyyntömäärien minimointi

Suurin yksittäinen tekijä verkkosivun latautumisajassa ovat HTTP-pyynnöt. Ne vievät siitä keskimäärin jopa 80 %. Tämän vuoksi HTTP-pyyntömäärien minimointi on ensimmäinen asia, mitä web-sovelluksen optimoimiseksi kannattaa tehdä. (Best Practices for Speeding Up Your Web Site.)

Yhdellä HTML-sivulla saattaa olla viitteitä useisiin eri CSS- ja Javascript-tiedostoihin. Näille kaikille pitää tehdä omat HTTP-pyyntönsä. Pyyntöjen määrää on helppo vähentää merkittävästi yhdistämällä kaikki Javascript-koodi samaan js-tiedostoon ja vastaavasti kaikki CSS-tyylit yhteen CSS-tiedostoon. Tämän jälkeen HTML-sivulla ei tarvitse tehdä kuin yhdet viittaukset näihin yhdistelmätiedostoihin. (Smits 2014.)

Myös kuvia saattaa yhdellä sivulla olla useita, ja niistäkin jokaista varten täytyy tehdä oma HTTP-pyyntö. Tähänkin sopii saman tyylinen ratkaisumalli kuin CSS:n ja JS:n tapauksessa. Kuvat voi sisällyttää samaan kuvatiedostoon ja näyttää siitä CSS:n avulla tarvittava osa. Tämä on kätevä tyyli esimerkiksi erilaisten pienten kuvakkeiden ja ikonien näyttämiseen. (Smits 2014.)

2.4 HTML-, CSS- ja Javascript-tiedostojen minifiointi

HTML-, CSS- ja Javascript-tiedostot sisältävät niiden suorittamisen kannalta paljon ylimääräistä tavaraa. Poistamalla kaikki kommentit, sisennykset ja rivinvaihdot saadaan pienennettyä tiedostojen kokoa sekä samalla tehostetaan niiden ajamista. (Smits 2014.)

Tiedostojen minifiointi onnistuu helposti esimerkiksi valmiiden sovellusten avulla.

Tällaisia ovat mm. Minify (<https://github.com/mrclay/minify>), JSCompress (<http://jscompress.com/>) ja CSS Compressor (<http://csscompressor.com/>).

2.5 Kuvien optimointi

2.5.1 Kuvien korvaaminen

Lisättäessä kuvaa web-sovellukseen tai –sivulle kannattaa ensimmäiseksi miettiä, tarvitseeko se lisätä nimenomaan kuvamuodossa. Yksinkertaisempia kuvia ja kuvioita voi luoda CSS:n avulla, jolloin säästetään resursseja kuvien lataamiselta. CSS:n avulla kuville pystyy luomaan myös erilaisia tehokeinoja kuten varjostuksia sekä liukuvärejä. (Grigorik n.d.)

Joissain tapauksissa kuvia voi korvata web-fonttien avulla. Tällöin pystyy helposti välttämään skaalaukseen liittyvät ongelmat. Yleisesti ottaen tekstisisältöjen esittämistä kuvamuodossa tulisi pyrkiä välttämään, jos ne on mahdollista näyttää tekstinä. Sen lisäksi että kuvassa oleva teksti lisää ylimääräistä lataamista se myös laskee käyttäjäystävällisyyttä. Kuvassa olevaa tekstiä ei voi valita ("maalata") tai hakea eikä siihen muutenkaan pääse käsiksi. (Grigorik n.d.b)

2.5.2 Oikean kuvaformaatin valinta

Sama yksittäinen kuvaformaatti ei ole optimaalisin jokaisessa tapauksessa vaan oikea formaatti tulee valita tilanteen mukaan. Yleisimmät käytössä olevat ja kaikilla selaimilla toimivat kuvaformatit ovat PNG, JPG ja GIF. Näillä kaikilla on omat erityisominaisuutensa (ks. taulukko 1.). (Grigorik n.d.b)

Taulukko 1. Kuvaformaattien ominaisuudet

Formaatti	Läpinäkyvyys	Animaatio	Selaintuki
PNG	Kyllä	Ei	Kaikki
JPG	Ei	Ei	Kaikki
GIF	Kyllä	Kyllä	Kaikki

GIF on paras vaihtoehto, jos kuvassa täytyy olla animaatiota. Se on tällä hetkellä ainoa universaali kuvaformaatti, jolla tämä ominaisuus on tarjolla. GIF-formaatilla on käytössä enintään 256 väriä eikä sen pakkauskaan ole tehokkaimmasta päästä, joten liikkumattoman kuvan esittämiseen se ei ole optimaalisin vaihtoehto. On myös hyvä muistaa, että yksinkertaisempia animaatiota voidaan toteuttaa myös CSS:n ja JavaScriptin avulla. (Grigorik n.d.b)

PNG on oikea valinta, jos kuvassa tarvitsee olla läpinäkyvyyttä tai sen pitää olla erityisen tarkka. Kuvan tarkkuuden mukana tulee myös PNG:n huono puoli eli suuremmat tiedostokoot muihin verrattuna. PNG sopii hyvin esimerkiksi erilaisten kuvakkeiden, ikonien ja logojen formaatiksi. (Grigorik n.d.b)

Valokuvien, kuvakaappausten ja muiden vastaavien kuvien formaatiksi sopii parhaiten JPG. Se on hävikillisen ja hävikittömän pakkaamisen yhdistelmä, jolla kuvan koko saadaan selvästi PNG:tä pienemmäksi. JPG-kuvan laatua (tarkkuutta) voi myös säätää ja saada näin mahdollisimman hyvä koko/laatu –suhde. (Grigorik n.d.b)

2.5.3 Oikean kokoisen kuvan käyttäminen

Vaikka CSS:n avulla on helppo määrittää, minkä kokoisena (pituus ja leveys) kuva näytetään sivulla, tähän ei tulisi tyytyä. Hyvä nyrkkisääntö on muokata alkuperäinen kuva suoraan sen kokoiseksi, millaisena se aiotaan näyttää. Väärän kokoisen kuvan käyttäminen lisää ylimääräistä lataamista ylimääräisten pikselien muodossa sekä kuluttaa CPU-resursseja skaalaamiseen. (Grigorik n.d.b)

2.5.4 Kuvaoptimointisovellusten hyödyntäminen

Kuvien optimointia varten on olemassa useita ilmaisia sovelluksia, jotka karsivat kuvista kaiken ylimääräisen pitäen kuvan laadun kuitenkin hyvänä. Tällaisia sovelluksia on tarjolla niin omalle tietokoneella ladattavina kuin myös suoraan selaimessa toimivina. Hyvä suoraan selaimella käytettävä kuvan optimointisovellus on esimerkiksi Optimizilla (<http://optimizilla.com/>). (Grigorik n.d.b)

2.5.5 Vektorigrafiikan käyttäminen

Perinteisten kuvaformaattien lisäksi nykyään on yleistynyt vektorigrafiikan hyödyntäminen kuvien esittämiseksi (SVG-formaatti). Tämä sopii erityisen hyvin yksinkertaisempien kuvien esittämiseen (logot, symbolit, ikonit jne.). Vektorigrafiikan ehdoton etu on sen rajattomassa skaalautuvuudessa. Samalla SVG-tiedostolla voi esittää ison ja pienen kuvan tarkkuuden kärsimättä. Monimutkaisemmille kuville kuten valokuville vektorigrafiikka ei kuitenkaan kunnolla sovellu. (Grigorik n.d.b)

2.6 Selaimen välimuistin hyödyntäminen

Idea Internet-selaimen välimuistin takana on selkeä: tallennetaan tiedostoja paikallisesti, jotta niitä ei tarvitse jokaisella sivun vierailukerralla ladata uudestaan. Käytännössä siis kun käyttäjä avaa ensimmäistä kertaa uuden verkkosivun, selain joutuu lataamaan kaiken sisällön palvelimelta. Samalla selain tallentaa osan sisällöstä omaan välimuistiinsa. Kun käyttäjä sitten vierailee kyseisellä sivulla uudelleen samalla selaimella, sivun sisältö latautuu nopeammin osan löytyessä jo valmiiksi paikalliselta koneelta. Usein tämä nopeusero ensimmäisen ja toisen latauskerran välillä on silminnähtävä. (Grigorik n.d.a)

Ongelma välimuistin kanssa tulee, jos tiedoston sisältö muuttuu palvelimella, mutta selain jatkaa vanhan version näyttämistä. Tiedostojen versiopäivityksiin voi hyödyntää ETag-, Expires- ja Max-Age -määrittelyksiä. (Grigorik n.d.a)

ETag-sormenjälki on tiedostolle muokkauksen yhteydessä luotava yksilöllinen tunnistekoodi. Selain kysyy ensin palvelimelta tiedoston ETagin ja vertaa sitä vastaavaan

välimuistissa olevaan. Jos Etagit eivät täsmää, on palvelimella oleva versio päivittynyt ja se ladataan selaimelle. (Grigorik n.d.a)

Expires-määrittäminen on nimensä mukaisesti tiedostolle annettava viimeinen käyttöpäivä. Selain katsoo ensin, vieläkö on aikaa jäljellä ennen Expires-päiväystä ja vain jos sitä ei ole, se lähettää palvelimelle pyynnön uudesta versiosta. Tällä keinolla pystytään välttämään ylimääräiset kyselyt palvelimen suuntaan kokonaan, jos tiedetään, ettei tiedosto päivity ennen jotain tiettyä päivämäärää. (Grigorik n.d.a)

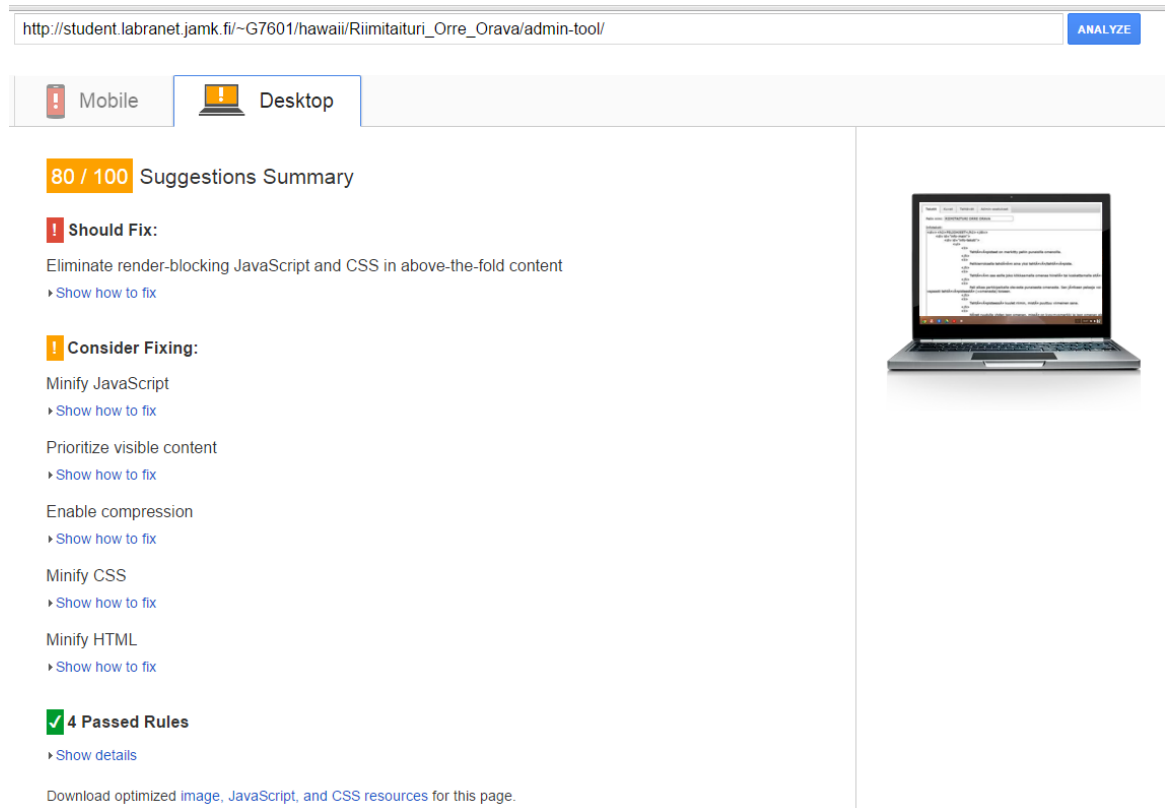
Max-Age-määrittäminen on käytännössä sama asia kuin Expires, mutta siinä annetaan päivämäärän sijaan sekunteina aikamäärä, jonka kuluttua tiedosto vanhenee. (Grigorik n.d.a)

2.7 Apuvälineet

Verkkosovellusten optimointia ja suorituskyvyn testaamista varten on olemassa useita ilmaisia apuvälineitä. Monet näistä osaavat antaa sivusta nopean palautteen, mitkä asiat ovat kunnossa ja missä olisi vielä parantamisen varaa. Luvuissa 2.7.1-4 esitellään muutama kätevä optimoinnin apuväline.

2.7.1 Google PageSpeed Insights

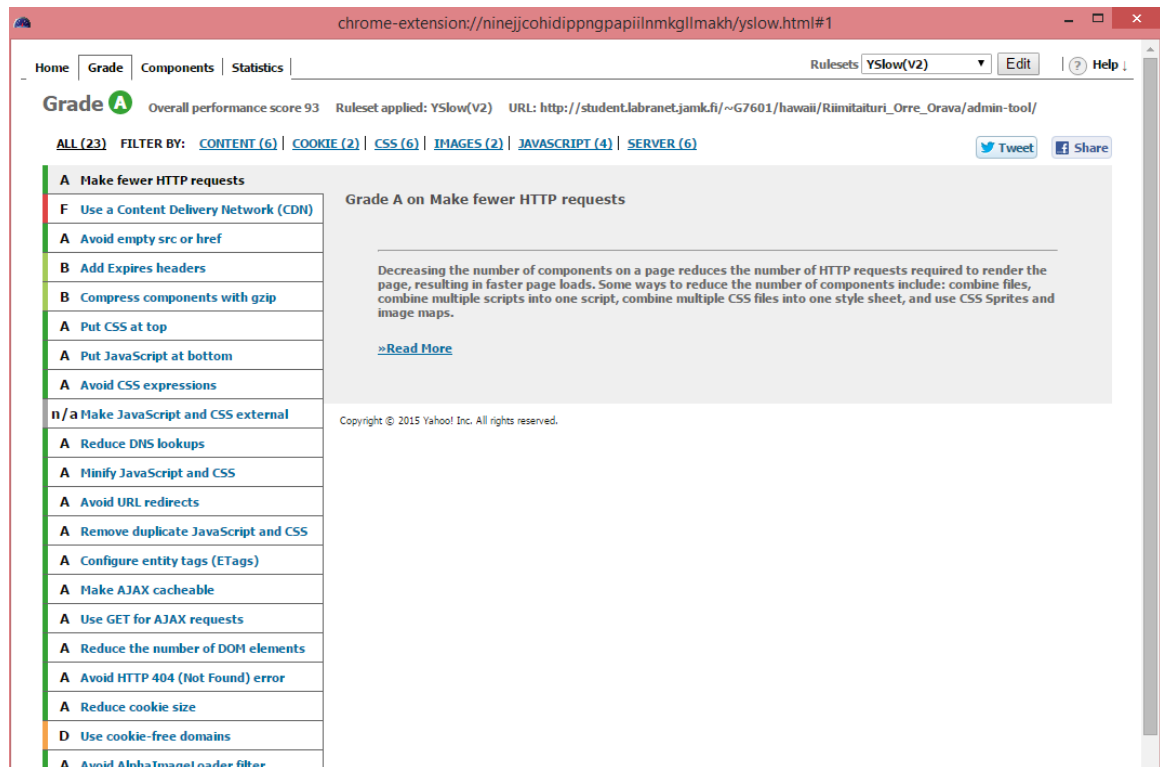
PageSpeed Insights (<https://developers.google.com/speed/pagespeed/insights/>) on Googlen tarjoama suoraan verkkoselaimella toimiva palvelu, joka analysoi käyttäjän antaman verkkosivun ja antaa sille korjausehdotukset. Sivulle syötetään testattavan sivun linkki, minkä jälkeen PageSeed Insights tekee analyysin ja näyttää tulokset käyttäjälle. Arvioitavat kohteet ovat nopeus erikseen mobiililaitteille ja pöytäkoneille sekä käyttäjäystävällisyys. Kaikki osa-alueet pisteytetään väliltä nollasta sataan. Sovellus antaa suorat tiedot asioista, jotka ovat kunnossa ja jotka vaativat korjausta. Tietojen yhteydessä on myös kuvaus ja linkki ohjeisiin, miten asia korjataan. Esimerkki sovelluksen antamasta tulossivusta on nähtävillä kuviossa 5.



Kuvio 5. Kuvakaappaus PageSpeed Insights –analyysin tulossivusta

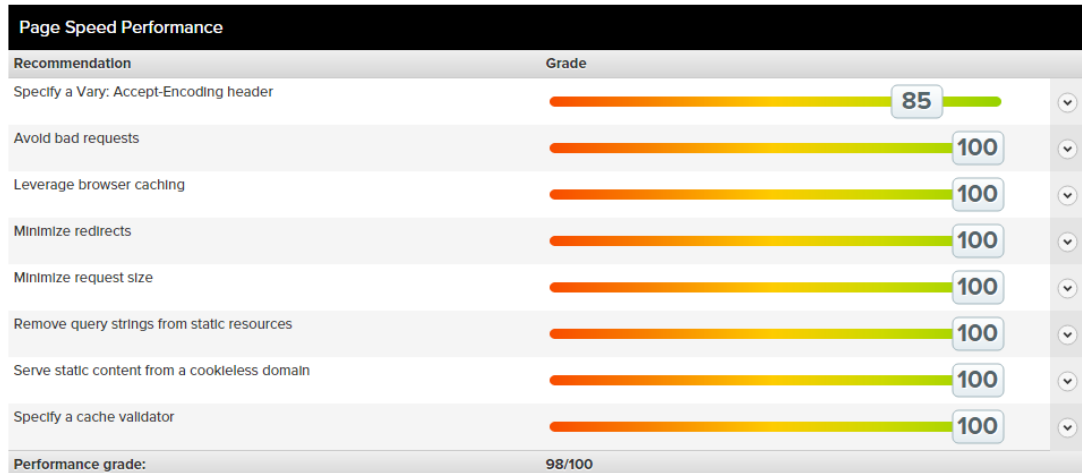
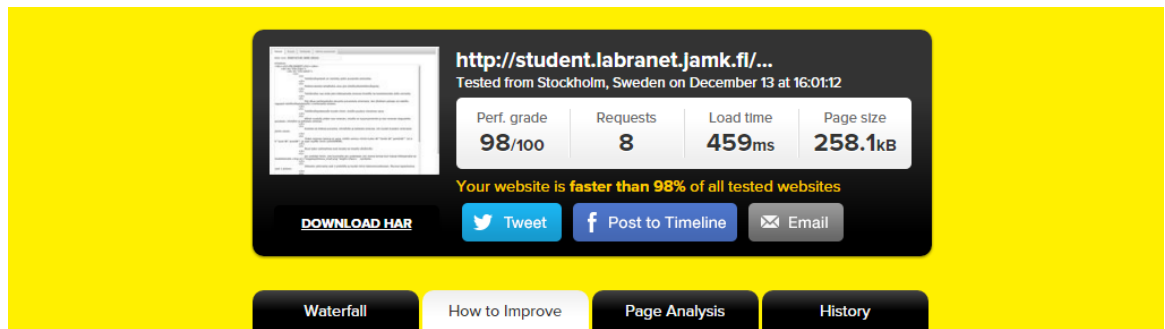
2.7.2 Yahoo!n YSlow

YSlow (<http://yslow.org/>) on selaimelle asennettava lisäosa, jonka avulla on helppo analysoida verkkosivujen suorituskyyä. Lisäosa on saatavilla mm. Chromelle, Firefoxille, Safarille ja Operalle. Kun se on asennettu, voi käyttäjä ajaa suorituskyytestin YSlow-kuvakkeesta suoraan sille sivulle, missä sillä hetkellä selaimellaan on. Testin jälkeen YSlow antaa reilusta 20 osa-alueesta erikseen arvosanan A-F ja antaa PageSpeed Insightsin tapaan korjausohjeet. Lisäksi YSlow antaa myös tilastot eri resurssien latauskoista ja niiden suhteellisista määristä. Kuviossa 6. on esitetty YSlow:n antama tulossivu.



Kuvio 6. Kuvakaappaus YSlow-analyysin tulossivusta

Pingdom Tools (<http://tools.pingdom.com/fpt/>) on myös samantyylinen analysointisovellus kuin kaksi aiemmin mainittua. Se toimii suoraan verkkoselaimella ilman erillisiä asennuksia. Sille annetaan testattavan sivun URL ja testin jälkeen se tarjoaa tuloksina erilaisia tilastotietoja. Se mm. antaa yleisarvosanan (0-100), kertoo kauanko sivun lataamiseen meni aikaa, paljonko pyyntöjä tehtiin ja paljonko niiden koko oli, esittää kaikkien pyydettyjen resurssien latausajat aikajanalla sekä kertoo eri resurssityyppien osuudet. Arvosanat (0-100) annetaan myös kahdeksalta eri osa-alueelta erikseen ja näihin annetaan korjausehdotukset. Sovellus myös säilyttää samalle sivulle tehdyt aiemmat testit tallessa ja helpottaa näin kehityksen seuraamista. Historiasta löytyvät myös valmiit diagrammigrafiikat testin yhteydessä. Kuviossa 7. on esitetty kuvakaappaus Pingdom Tools:n tulossivusta.



Kuvio 7. Kuvakaappaus Pingdom Tools –analyysin tulossivusta

2.7.3 Verkkoselainten omat kehitystyökalut

Myös useaan verkkoselaimeen (kuten Chrome ja Firefox) oletuksena kuuluvalla kehittäjän työkalulla pystyy tekemään omia suorituskykytestejä tai ainakin seuraamaan ja tallentamaan verkkoliikennettä ja latausaikoja sivulla. Javascriptissä olevat virheet on helpointa löytää kehitystyökalun ”Console”-välilehdelle tulevista logeista.

3 RIIMITAITURI ORRE ORAVAN

SUORITUSKYKYOPTIMOINTI JA BUGIKORJAUKSET

3.1 Ääniin liittyvien ongelmien korjaaminen

Äänien kanssa esiintyi paikoin kolmea erilaista vikaa: päällekkäin kuulumista, ajoitusongelmia sekä äänien kuulumattomuutta. Tehtäviin liittyviä ääniä oli mahdollista kuunnella samaan aikaan ohjeistusäänien kanssa. Jos klikkasi ensimmäistä rastiomenaa alkuohjeistusten ollessa kesken, rastisiirtymä-ääni alkoi soida ohjeistusten päälle. Tämä oli helppo korjata lisäämällä rastisiirtymä-funktion alkuun pysäytyskomento ohjeistusäänelle (`tutorialAudio1.pause();`). Vastaava ongelma oli myös ensimmäisen tehtävärastin ohjeistuksen aikaan, jos käyttäjä klikkasi ”Kuuntele uudestaan” – painiketta. Tämä ongelma ratkaistiin sillä, että tehtäväaudiolle annetaan lähdetiedosto vasta ohjeistuksen loputtua. Ennen sitä käyttäjä voi siis kuunnella vain ”tyhjää” tehtävänantoa.

Ohjeistusäänissä todettiin paikoittain olevan liian pitkiä hiljaisia taukoja. Nämä korjattiin yksinkertaisesti suoraan muokkaamalla ääniraidasta tyhjää pois. Äänien kuulumattomuusongelmat puolestaan liittyivät tablet-laitteisiin (erityisesti iPadiin), ja niitä käsitellään lisää luvussa 3.2.

3.2 Pätkivän palkintovideon korjaaminen

Joillakin laitteilla ja hitaammilla internet-yhteyksillä pelikierroksen lopussa näytettävät palkintovideot pätkivät jonkin verran. Suurin syy tähän oli se, että näytettävä video kolmesta eri vaihtoehdosta ladattiin vasta siinä vaiheessa, kun kierros loppui ja video piti näyttää. Tällöin video ei ollut lainkaan kerennyt latautua etukäteen ennen näyttämistä. Ratkaisuna tehtiin logiikkamuutos videon lähteen muuttamiseen. Uudessa mallissa palkintovideon lähdetiedosto vaihdetaan heti, kun tietty pisteraja ylittyy. Näin video kerkeää latautumaan rauhassa taustalla ennen kuin se näytetään kierroksen lopussa pelaajalle.

3.3 iPad-ongelmien korjaaminen

IPadilla ja mahdollisesti myös joillakin muilla tablet-laitteilla esiintyi joitakin ääniin ja videoihin liittyviä ongelmia. Rastien siirtymäriimit eivät kuuluneet, kierroksen lopussa ei tullut palkintopallivideota ja lisäksi peli vielä jumittui toisen kierroksen ensimmäiseen tehtävään. Syyksi näihin kaikkiin osoittautui sama ongelma, johon oli törmätty jo kevään aikana. Tablet-laitteille on Javascriptin play-toiminto video- ja audioelementeille oletuksena estetty. Tämä toiminto sallitaan ainoastaan tilanteissa, joissa käyttäjä itse aiheuttaa sen klikkaamalla tai jollain muulla omalla toiminnallaan. Lisäksi tämän ensimmäisen klikkauksen aiheuttaman soittamisen jälkeen play-toiminto toimii jatkossa samalle elementille myös ilman klikkausta. Tätä tietoa hyödynnettiin ongelman ratkaisussa.

Kaikki pelissä toistettavat video- ja audioelementit alustetaan sovelluksen alussa tyhjiksi eli niille ei anneta lähdetiedostoa. Kun käyttäjä aloittaa pelin klikkaamalla ”Jatka peliä”- tai ”Uusi peli”-painiketta, toistetaan heti aluksi kaikki nuo tyhjät elementit. Tämä vapauttaa elementit play-estosta, joten niille voidaan jatkossa lisätä lähdetiedosto ja niitä pystytään vapaasti toistamaan tarvittavissa osioissa.

3.4 Javascript-tiedostojen minimointi ja yhdistäminen

Pelissä käytettävät Javascript-tiedostot minifioitiin ja osa niistä yhdistettiin JSCompressin (<http://jscompress.com/>) avulla. Neljä useimmin muokattavaa JS-tiedostoa (main.js, addingDivs.js, tasks.js ja shop.js) päätettiin jättää omiksi tiedostoiksi ilman yhdistelyä, mutta niistäkin tehtiin minifioidut versiot. Kaikki ”ei muokattavat” tiedostot eli käytännössä kaikki käytössä olevat eri JS-kirjastot yhdistettiin ja minifioitiin yhdeksi libraries.js-tiedostoksi. Näin saatiin vähennettyä HTTP-pyyntöjen määrää seitsemällä.

Suurin osa JS-kirjastoista oli jo valmiiksi minimoituina versioina, joten niiden minimoimisesta ei lopulta erityisen isoa hyötyä saatu. Toisaalta muut JS-tiedostot olivat

sen verran pieniä jo valmiiksi, ettei niistäkään suurta säästöä saatu tavumäärinä, vaikka prosentuaalisesti muutos olikin huomattava.

Pelin HTML-tiedosto (index.html) ja CSS-tiedosto (styles.css) olivat sen verran pieniä, ettei niiden minimoinnille nähty tarvetta. Samalla niiden muokattavuus säilyi helppona.

Koko sovelluksen suurin ongelma suorituskyyvylle olivat 3D-mallit sisältävät JS-tiedostot, jotka ovat käytännössä vain JS-tiedostoiksi nimettyjä JSON-tiedostoja. Näitä tiedostoja on monta, sillä teknisistä syistä ne piti tehdä mm. jokaiselle eri huoneelle erikseen. Sen lisäksi nämä tiedostot ovat todella isoja. Ne oli tehty melko tiiviiksi jo valmiiksi, joten niiden minimoiminen ei tuonut käytännössä mitään etua. Tiedostoja ei voinut yhdistää yhdeksi jättitiedostoksi, sillä sellaisen lataaminen Threejs:ssä kaatoi tehokkaammankin koneen selaimen. Tämä todettiin jo viime keväänä isoimpien huonemallien kanssa ja niitä jouduttiin jo silloin keventämään mallinnuksen puolella.

Kaikkiaan minimoinnin avulla onnistuttiin tiedostojen kilotavumäärää vähentämään 63:lla eli kahdeksalla prosentilla. Tarkempi erittely muutoksista on esitetty taulukossa 2.

Taulukko 2. JS-tiedostojen viemän muistin määrä ennen ja jälkeen minimoinnin

Tiedosto	Koko ennen minimointia (KB)	Koko jälkeen minimoinnin (KB)	Erotus (KB)	Muutos prosentteina
addingDivs.js	17	11	6	-35 %
main.js	16	8	8	-50 %
tasks.js	19	10	9	-47 %
shop.js	7	5	2	-29 %
Kirjastotiedostot yhteensä	757	719	38	-5 %
Yhteensä	816	753	63	-8 %

3.5 Kuvien optimointi

Pelissä käytetään paljon kuvia valikoissa, painikkeissa, palkintokaupassa ja muuten käyttöliittymän yhteydessä, mutta vielä enemmän eri kuvatiedostoja on mukana 3D-ympäristön pintamateriaaleina. Yhteensä pelissä on jopa noin 170 kuvatiedostoa, joten niiden optimoinnilla voitiin tehdä jo varsin huomattavat erot ladattavien bittien määriin. Lähes kaikki kuvat olivat JPEG- tai PNG-muodossa. Ainoastaan pelin alussa näytettävän latausikkunan animoitu kuva on GIF-tiedosto.

Kaikki kuvat optimoitiin automaattisesti ilmaisen verkossa toimivan Optimizilla-sovelluksen avulla (<http://optimizilla.com/>). Sovellukseen ladattiin kuvat, minkä jälkeen se prosessoivat ne läpi. Uudelleen pakatut kuvatiedostot sai ladattua itselleen ZIP-paketissa. Kuviin ei tullut silminhavaittavia muutoksia, mutta niiden kokoa saatiin pienennettyä huomattavasti. Yhteensä kuvien viemää muistitilaa saatiin pienennettyä jopa lähes 70 % (ks. taulukko 3.).

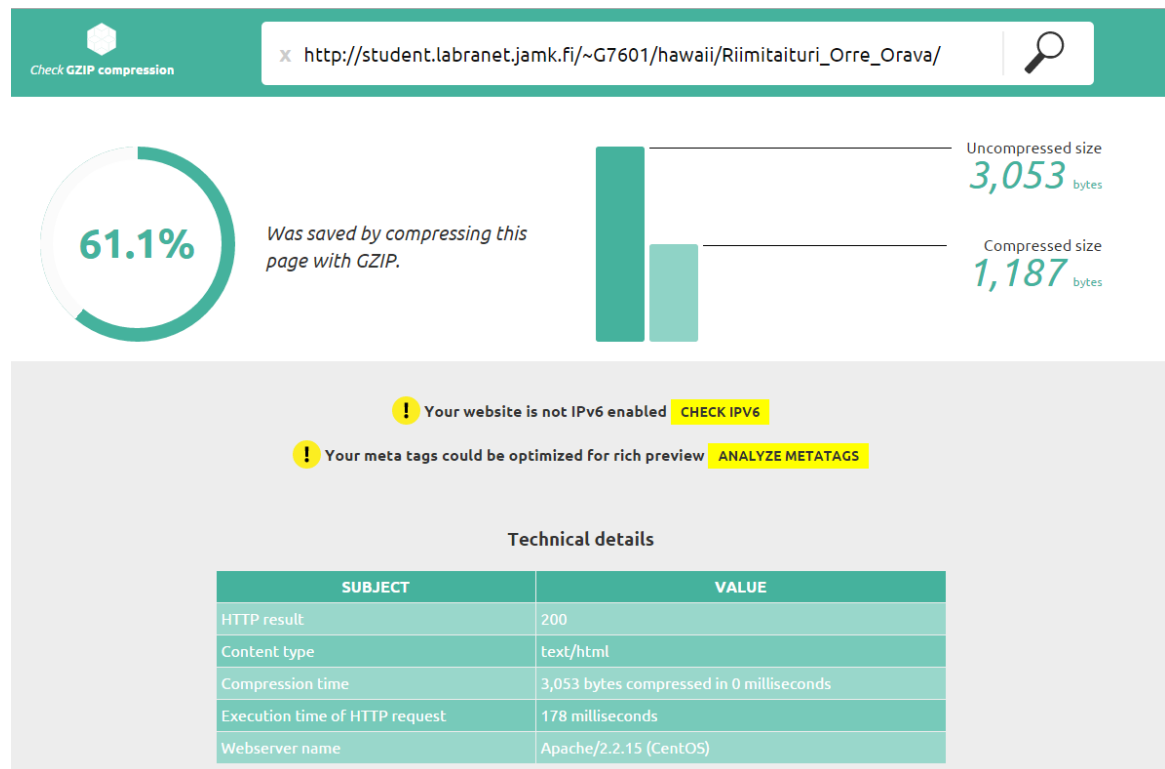
Taulukko 3. Kuvien viemän muistin määrä ennen ja jälkeen optimoinnin

Kuvien sijainti	Koko ennen optimointia (KB)	Koko jälkeen optimoinnin (KB)	Erotus (KB)	Muutos prosentteina
/img	1 330	789	541	-41 %
/img/kauppa	405	128	277	-68 %
/models	1 710	706	1004	-59 %
/models/Floor1	2 720	941	1779	-65 %
/models/rooms	3 370	418	2952	-88 %
/models/symbolit1	171	68	103	-60 %
/models/symbolit2	82	30	52	-63 %
Yhteensä	9788	3080	6708	-69 %

3.6 Palvelinpuolen optimointi

Palvelimelle tehtäviä GZIP-pakkauksen ja välimuistin hyödyntämisasetuksia ei voida ladattavaan pelipakettiin lisätä valmiiksi vaan ne on tehtävä uudelle palvelimelle erikseen. Onervalla näyttää GZIP-pakkaus olevan käytössä jo valmiiksi. Tämä todettiin testaamalla sivujen onerva.fi ja onetti.fi GZIP-pakkaus.

GZIP-pakkaamisen käyttöönottoa testattiin kuitenkin myös JAMK:n student-palvelimella. Se saatiin toimimaan kahdellakin eri tavalla lisäämällä tarvittavat määritykset .htaccess-tiedostoon. GZIP-testaussivu (<http://checkgzipcompression.com/>) ilmoitti pakkauksen säästäneen 61,1 % latauksen koosta (ks kuvio 8).



Kuvio 8. GZIP-pakkaustestin tulossivu

Myös tiedostojen välimuistisäännöt pitää tehdä palvelimella erikseen, joten niitäkään ei valmiiseen pelipakettiin saatu lisättyä. Toisaalta ainakaan kaikkia tiedostoja ei välttämättä halutakaan pitkäksi aikaa välimuistiin, jos niitä (esim. kuvia) halutaan muokata ja vaihtaa.

3.7 Testitulokset

Sovellusta testattiin projektin aikana Google PageSpeed Insightsillä, YSlow:lla ja Pingdom Toolsilla. Kaikki antoivat samansuuntaisia parannuskehotuksia. Ennen korjauksia tehdyissä testeissä kehoitettiin ottamamaan käyttöön GZIP-pakkaus, säätämään palvelimen välimuistisäännöt, vaihtamaan JS- ja CSS-tiedostojen lähdeviittausten paikkaa HTML-sivulla, minimoimaan JS-tiedostot ja optimoimaan kuvat.

Kuten luvussa 3.5 kerrotaan, otettiin student-palvelimella GZIP-pakkaaminen onnistuneesti käyttöön. Jostain syystä PageSpeed Insights ja YSlow valittivat senkin jälkeen testituloksissa, ettei tämä ominaisuus olisi käytössä. Ilmeisesti toiminto pitäisi aktivoida jollakin tietyllä tavalla, jotta nämä testisivut havaitsisivat aktivoinnin. Nopean googlauksen antamat tavat eivät kuitenkaan toimineet student-palvelimella.

JS-tiedostojen lähdeviittausten paikasta tuli huomautus sen takia, koska ne pysäyttivät turhaan muun sivun lataamisen. Näiden paikkaa kokeiltiin vaihtaa ehdotusten mukaisesti, mutta tällöin sovellus ei toiminut ja konsoliin tuli virheilmoituksia. Parempi oli siis säilyttää alkuperäinen järjestys.

3.8 Raskaiden 3D-mallien ongelma

Luvuissa 3.3 – 3.5 mainitut optimointikeinot vaikuttavat lähinnä sivun latausaikaan eivätkä niinkään siihen, kuinka tehokkaasti ja sujuvasti peli toimii alkulatausten jälkeen. Suurimmat ongelmat sovelluksessa varsinkin vanhemmille ja tehottomammille laitteille aiheutuvat kuitenkin käytetystä varsin uudesta three.js-tekniikasta. Tarkemmin sanottuna three.js-3D-mallien suuri koko ja määrä on sovelluksen ”pullonkaulaongelma”.

Pelissä käytettäviä 3D-malleja yksinkertaistettiin ja niiden kokoa pienennettiin huomattavan paljon jo edellisen projektin aikana. Nykyiset mallit on siis jo optimoitu varsin hyvin, eikä niitä alettu enää tämän projektin puitteissa yksinkertaistamaan. Myös mallien JS-tiedostot olivat jo valmiiksi käytännössä minimoituja, joten sitäkään kautta ei apuja ollut saatavilla.

Malleissa käytettyjen pintatekstuurien lähdekuvien kokoja sai pienennettyä huomattavasti luvussa 3.4 tehdyn kuvaoptimoinnin avulla.

Lopulta ongelmaan ei löytynyt mitään järkevää ratkaisua. Käytössä olleista testilaitteista peliä ei pystynyt pelaamaan iPad2:lla, mutta mm. iPad Airilla, Samsung Galaxy S5:llä ja useammalla kannettavalla tietokoneella sekä pöytätietokoneella peli toimi hyvin.

iPad2:lla suoritettiin testejä vähentämällä peliin ladattavien mallien määrää. Eri yhdistelmistä saatiin seuraavia tuloksia:

Tehtäväomenat + 5 tehtävää + kaikki kerrokset + landscape + symbolit -> ei toimi

Tehtäväomenat + 5 tehtävää + kaikki kerrokset + landscape -> ei toimi

Tehtäväomenat + 5 tehtävää + 2 kerrosta + landscape -> ei toimi

Tehtäväomenat + 1 tehtävä -> toimii

Tehtäväomenat + 1 tehtävä + 1. kerros -> toimii

Tehtäväomenat + 1 tehtävä + 1. kerros + landscape -> toimii

Tehtäväomenat + 5 tehtävää + 1. kerros + landscape -> toimii kohtuullisesti

Tehtäväomenat + 7 tehtävää + 1. kerros + landscape -> toimii huonosti

Tehtäväomenat + 10 tehtävää + 1. kerros + landscape -> toimii todella huonosti

Tulosten perusteella olisi siis käytännössä voitu tehdä iPad2:lle jonkinlainen erikoislogiikka, jolla ladattaisiin peliin ainoastaan ensimmäinen kerros, ympäristö ja noin viisi tehtävää. Tämä ratkaisu ei kuitenkaan tuntunut järkevälle useastakaan syystä. Ensinnäkin tällainen peli olisi ollut huomattavan paljon vajaampi verrattuna muille alustoille tarjottavaan peliin. Toisekseen vaadittavat huomattavat logiikkamuutokset olisivat aiheuttaneet suuren riskin uusille virheille myös muilla laitteilla. Uudet muutokset olisi pitänyt huomioida myös sisällöntuottamistyoäkalussa. Kaikkiaan olisi siis tarvinnut tehdä paljon ylimääräistä työtä kohtuullisella riskillä kuitenkin melko pienen hyödyn vuoksi (toimivuuden lisääminen yksittäiselle laitteelle ja sekin vajavaisena).

Keväällä oli mietitty WebGL:ää tukemattomille ja huonosti tukeville alustoille mahdollisuutta toteuttaa peli videosiirtymiä hyödyntäen. Tämä olisi ollut kuitenkin huomattavan kömpelö ratkaisu, joka olisi aiheuttanut paljon ylimääräistä työtä ja vaikeuttanut pelin jatkokehitystä oleellisesti. Kaikki siirtymät olisi pitänyt erikseen nauhoittaa ja lisätä niiden lähteet koodiin. Lukuisat videoklipit luonnollisesti olisivat myös vaatineet paljon lisää muistia palvelimelta sekä aiheuttaneet huomattavan määrän ylimääräistä ladattavaa sovellukselle.

4 SOVELLUKSEN JATKOKEHITYS

4.1 Pienet lisätoiminnot

4.1.1 Alkuvideon ohitusmahdollisuus

Käyttäjän aloittaessa uuden pelin näytetään hänelle aina pelin alkuvideo. Video on reilun minuutin mittainen, joten muutaman pelikerran jälkeen sen pakollinen kokonaan katsominen voi alkaa häiritä käyttökokemusta. Siispä tähän haluttiin tehdä videon ohitusmahdollisuus.

Tämä oli melko helppo ja nopea lisäys, joka toteutettiin videon alakulmassa näkyvällä ”Ohita alkuintro”-linkillä. Linkin onclick-funktioksi asetettiin sama `introEnd()`, joka ajetaan myös siinä tapauksessa, että video loppuu.

4.1.2 Tehtäväsiirtymän ohitusmahdollisuus

Tehtävärasteille siirryttäessä tulee aina rastikohtainen siirtymä, jonka aikana kerrotaan myös paikkaan liittyvä esittelyruno. Malttamattomammalle pelaajalle nämä siirtymät saattavat käydä pitkästyttäväksi, joten myös tähän vaiheeseen haluttiin ohitusmahdollisuus.

Alkuvideon ohitusmahdollisuuteen verrattuna siirtymän ohitus oli hieman monimutkaisempi. Myös tähän tarkoitukseen tehtiin alakulmaan linkki, jota klikkaamalla

siirtymä nopeutetaan. Peruseriaate on, että kaikki muistissa (meneillään) olevat kamera-ajot poistetaan ja hypätään suoraan kamera-ajon viimeiseen vaiheeseen. Samaan aikaan esittelyriittelyn ääniraita pysäytetään ja sen pysäytyskohta siirretään takaisin ääniraidan alkuun.

4.1.3 Mobiililaitteilla pystyasennossa pelaamisen estäminen

Peli on suunniteltu pelattavaksi vaaka-asennossa olevalla näytöllä. Tästä syystä pelin sisältö skaalautuu huonosti pystyasennossa olevalle näytölle. Ensimmäinen korjausajatus tähän ongelmaan oli pakottaa sisällön näyttäminen aina vaaka-asentoon. Tähän ei kuitenkaan löytynyt toimivaa tekniikkaa, joten oli pakko keksiä jotain muuta.

Lopulta päädyttiin ratkaisuun, jossa pelin sisältö piilotetaan aina pelaajan kääntäessä näytön pystyasentoon ja sen sijaan näytetään ilmoitus, että peliä voi pelata vain vaaka-asennossa. Tämä onnistui helposti CSS:n avulla hyödyntämällä näyttökohtaisia sääntöjä ”@media only screen and (orientation:portrait)” ja ”@media only screen and (orientation:landscape)”.

Ongelma ei kuitenkaan ratkennut kokonaan pelkästään tuolla keinolla. Edelleen jos pelaaja avasi sivun laitteen ollessa pystyasennossa, latautui pelin sisältö alussa taustalla väärin, ja pelaajan kääntäessä laitteensa oikeaan asentoon tuli jälleen huonosti skaalautunut sisältö näkyviin. Tämä korjattiin lisäämällä näyttämällä näissä tapauksissa ilmoitus, jossa kehoitetaan ensin kääntämään näyttö vaaka-asentoon ja sitten lataamaan sivu uudestaan. Tämä viesti ei lähde pois näkyvistä ennen kuin sivu on ladattu uudestaan.

4.2 Palkintokauppa

Peliin oli suunnitteilla tehdä jonkinlainen palkintokauppa jo ”Projekti”-opintojaksolla, mutta lopulta sen toteuttamiseen ei riittänyt aikaa. Ideana oli siis, että pelaaja voisi ansaitsemillaan omenoilla ostaa palkintoja pelin päähahmolle Orrelle. Grafiikat tätä varten oli kuitenkin tehty jo valmiiksi Illustrator-tiedostoon.

Palkintokauppa on kuva Orresta rantamaisemassa. Tähän kuvaan voi ostaa lisää erilaisia esineitä ja asioita omenapisteitä käyttäen. Näkymä ilman yhtään palkintoa on esitetty kuviossa 9.



Kuvio 9. Kaupan pohjakuva ilman palkintoja

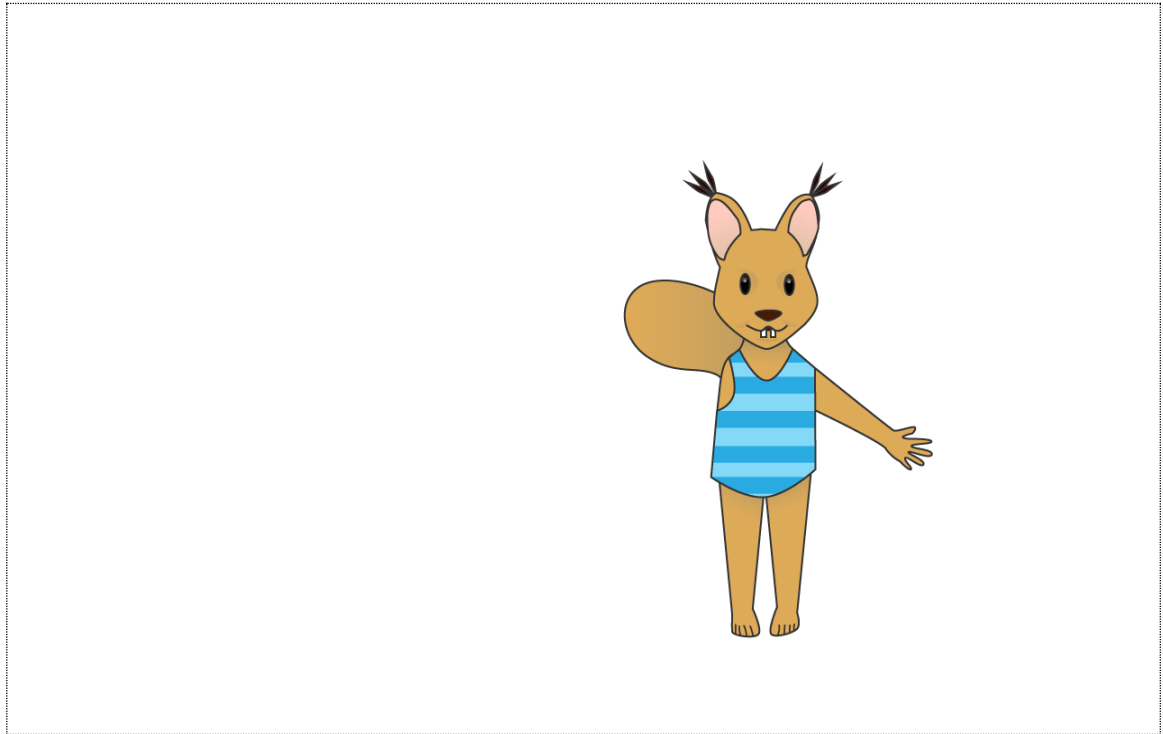
Palkintoja on tarjolla yhteensä kymmenen kappaletta. Nämä ovat laiva, aurinko, mökki, hattu, kukat, rantapallo, aurinkolasit, lainelauta, riippumatto ja shortsit. Palkintokauppanäkymä kaikkine palkintoineen on esitetty kuviossa 10.



Kuvio 10. Kauppa kaikkien palkintojen kanssa

Ensimmäinen haaste oli keksiä, miten saavutetut palkintokuvat saadaan lisättyä valmiiseen taustakuvaan oikeille paikoilleen. Osa kuvista piti saada kutakuinkin pikselillään oikeaan paikkaan (esim. aurinkolasit). Lisäksi joidenkin palkintokuvien päällekkäisyysjärjestyksellä oli väliä ja osa piti jopa saada taustakuvan tiettyjen elementtien väliin (esim. laiva piti saada osittain kukkulan taakse, mutta kuitenkin veden eteen ja rantapallo piti saada Orren käden taakse mutta muun kehon eteen).

Ratkaisuna oli tehdä jokaisesta palkintokuvasta sekä taustakuvan eri kerroksista omat samanpituiset ja –levyiset osittain läpinäkyvät PNG-kuvat, joissa varsinainen kuva oli valmiiksi oikealla paikalla. Nämä kaikki kuvat asetettiin CSS-tyyliin avulla tarkalleen samalle kohdalle näytölle (`position: absolute;`) ja niiden päällekkäisyysjärjestys määriteltiin CSS:n `z-index`in avulla. Esimerkki tällaisesta kuvakerroksesta kuviossa 11.



Kuvio 11. Palkintokaupan yksittäinen kuvakerros: Orre ilman ”etukättä”

Helpomman muokattavuuden ja hallinnoinnin vuoksi palkintojen perustiedot on tallennettu JSON-tiedostoon. Tämä tiedosto sisältää tiedot palkintojen kuvien URL:sta, ID:stä ja niiden hinnoista.

Kun pelaaja ostaa kaupasta palkinnon, tallennetaan tästä tieto Javascriptissa taulukkomuuttujaan. Sama taulukko tallennetaan myös jokaisen muutoksen jälkeen pelaajan selaimen local storageen, jotta voitettut palkinnot säilyvät myös jatkettaessa vanhaa peliä.

Kauppa on toteutettu omana divinä, joka luodaan, kun käyttäjä klikkaa ensimmäistä kertaa ostoskärrypainiketta. Tässä vaiheessa sovellus tarkastaa muistista, onko pelaaja jo voittanut palkintoja ja lisää niiden kuvat näytölle. Voittamattomille palkinnoille luodaan omat painikkeet. Kun kauppa-div on kerran luotu, jatkossa se vain piilotetaan näkyvistä CSS:n avulla käyttäjän poistuessa kaupasta ja näytetään uudelleen käyttäjän palatessa sinne. Kuvakaappaus pelin palkintokaupasta näkyy kuviossa 12.



Kuvio 12. Pelin kaupanäkymä

Palkintojen ostaminen luonnollisesti vähentää pelaajan keräämien pisteomenoiden määrää. Kierroksen lopussa näytettävä palkintovideo kuitenkin näytetään kerättyjen pisteiden mukaan, joten peliin tarvittiin yksi pistemuuttuja lisää. Siispä pelissä on nyt omat muuttujansa ostamiseen käytettävien omenoiden määrälle (näkyvä käyttäjälle) sekä pelin aikana kerätylle kokonaispistemäärälle, jonka perusteella näytetään palkintovideo.

4.3 Työkalu uuden sisällön tuottamiseen

Pelin sisältöjen muokattavuuden helpottamiseksi sekä valmiin pelipohjan jatkohyödyntämistä varten peliin luotiin kokonaan uusi työkalu uuden sisällön tuottamiseen ja vanhan muokkaamiseen. Ideana siis oli suoraan selaimella käytettävä työkalu, jolla pystyy muokkaamaan pelin kuva-, teksti-, ääni- ja videosisältöjä sekä valitsemaan olemassa olevista tehtävärastipaikoista käyttöön otettavat. Tämän avulla voisi siis jopa luoda kokonaan uuden pelin toimimaan samalle pelimekaniikalle.

4.3.1 Perusrakenne ja käytetyt tekniikat

Kaikki sisällönhallintatyökalun tiedostot sijaitsevat ”admin-tool”-kansiossa. Ne ovat kokonaan oma kokonaisuutensa eivätkä ne itsessään vaikuta mitenkään itse peliin. Eli käytännössä ”admin-tool”-kansion voisi poistaa kokonaan eikä se vaikuttaisi pelin pelattavuuteen millään lailla.

Työkalu on toteutettu PHP:llä. Sen melko yksinkertainen välilehtimallinen käyttöliittymä on toteutettu jQueryn Tabs-kirjaston avulla (<https://jqueryui.com/tabs/>). Jokaiselle välilehdelle on oma PHP-tiedostonsa, joissa hoidetaan oma osa-alueensa. Lisäksi käytössä on muutama aputiedosto, joissa olevia funktioita saatetaan hyödyntää useammassa paikassa.

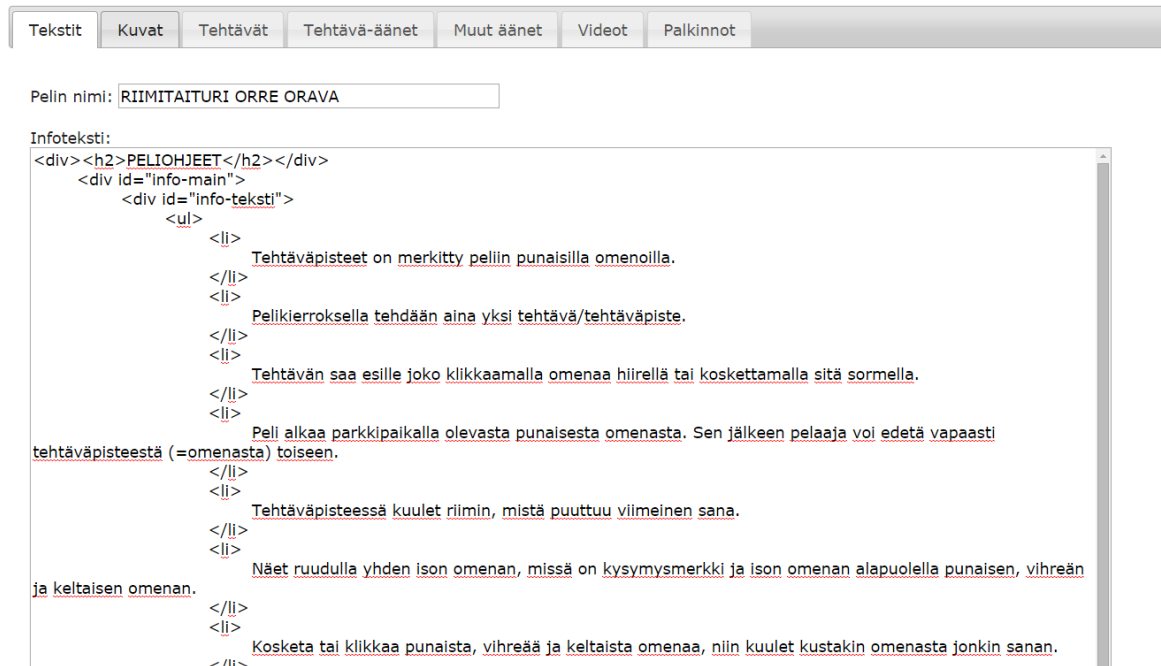
Suurin osa työkalun suorittamista tehtävistä on eri lähdetiedostojen (kuvat ja videot) korvaamista uusilla, mutta sillä myös muokataan tekstimuotoisten lähdetiedostojen sisältöjä (JSON, TXT). Jossain vaiheessa oli keskustelua, pitäisikö sovellukseen ottaa mukaan oikea tietokanta tiedon tallentamisen helpottamiseksi. Tähän ei kuitenkaan lopulta nähty tarvetta vaan JSON-tiedostot ajoivat asian riittävän hyvin. Tietokannan poisjättäminen myöskin helpottaa huomattavasti sovelluksen käyttöönottoa uusilla palvelimilla. JSON-tiedostoihin tallennettu tieto on heti käyttövalmiina ilman mitään erillisiä tietokannan perustamisia ja konfigurointeja.

4.3.2 Tekstien muokkaaminen

Muokkaustyökalun ensimmäisellä välilehdellä voi muokata pelin nimeä ja infotekstiä. Nämä tekstisisällöt on tallennettu ”text-contents”-kansioon omiin TXT-tiedostoihinsa. PHP lataa näiden tekstitiedostojen sisällöt esitetylle lomakkeelle, jossa käyttäjä voi tehdä niihin haluamansa muutokset. Jos käyttäjä tallentaa muutokset, PHP:ssä muokataan tekstitiedostoihin uusi sisältö. Pelissä pelin nimi ja infoteksti ladataan Javascriptin avulla tekstitiedostoista oikeisiin paikkoihin. Infoteksti kirjoitetaan HTML-muodossa. Kuvakaappaus ”Tekstit”-välilehdestä näkyy kuviossa 13.

Jonkin aikaa ongelmia aiheuttivat lomakkeella ”hajonneena” näkyneet ja korjattuna väärin tallentuneet ääkköset ja erikoismerkit. Tähän selvisi lopulta syyksi tekstin

enkoodauksen muuttuminen, joka oli aiheutunut suoraan WinSCP:n tekstieditorilla tallennetuista muutoksista PHP-tiedostossa. Enkoodaus vaihdettiin takaisin UTF-8:aan ja ongelma korjaantui.



Kuvio 13. ”Tekstit”-välilehti sisällönhallintatyökalussa

4.3.3 Kuvien muokkaaminen

Muokkaustyökalun ”Kuvat”-välilehdellä ovat lomakkeet useimpien pelissä käytettävien kuvien vaihtamiseen. Käyttäjä voi halutessaan vaihtaa mm. alkuvalikon painikkeiden, yleisnäkyvän valikon kuvakkeiden sekä tehtävien vastausten ja vastausalueen kuvat.

Kun käyttäjä on valinnut uuden kuvan ja tallentanut sen, tarkastetaan tiedoston sopivuus. Jos kaikki on kunnossa, tallennetaan kuva yksinkertaisesti edellisen käytössä olleen kuvan tilalle, jolloin se tulee automaattisesti käyttöön pelissä.

Kuviossa 14. näkyy kuvakaappaus ”Kuvat”-välilehdestä.

Tekstit	Kuvat	Tehtävät	Tehtävä-äänet	Muut äänet	Videot	Palkinn
---------	--------------	----------	---------------	------------	--------	---------

Alkuvalikko

"Jatka peliä" -painike (.png): Ei valittua tiedostoa

"Uusi peli" -painike (.png): Ei valittua tiedostoa

"Info" -painike (.png): Ei valittua tiedostoa

Yleisnäkymä

Info-painike (.png): Ei valittua tiedostoa

Pistemäärän taustakuva (.png): Ei valittua tiedostoa

Kauppa-painike (.png): Ei valittua tiedostoa

Tehtävärasti

Nimikyltti (.png): Ei valittua tiedostoa

Vastauksen pudotusalue (.png): Ei valittua tiedostoa

Vastausvaihtoehto 1 (.png): Ei valittua tiedostoa

Vastausvaihtoehto 2 (.png): Ei valittua tiedostoa

Kuvio 14. "Kuvat"-välilehti sisällönhallintatyökalussa

4.3.4 Tehtävärastien valinta

”Tehtävät”-välilehdellä käyttäjä voi valita, mitkä tehtävärastipaikat peliin otetaan mukaan. Tehtävärasteja on valittavissa yhteensä 22 ja niitä voi valita haluamansa määrän. Lisäksi samalla voi määritellä paikan nimen (näkyv pelattaessa tehtävärastilla nimikyltissä) ja siirtymäajan paikalle siirtymiselle. Siirtymäajan määrittäminen oli alunperin tarkoitus määrittää Javascriptissä automaattisesti siirtymä-äänien pituuden mukaan, mutta tämä jätettiin lopulta pois teknisten ongelmien takia. Tätä mahdollisuutta ei enää ehditty tutkimaan pidemmälle tämänkään projektin puitteissa. Kuvakaappaus ”Tehtävät”-välilehdestä näkyy kuviossa 15.

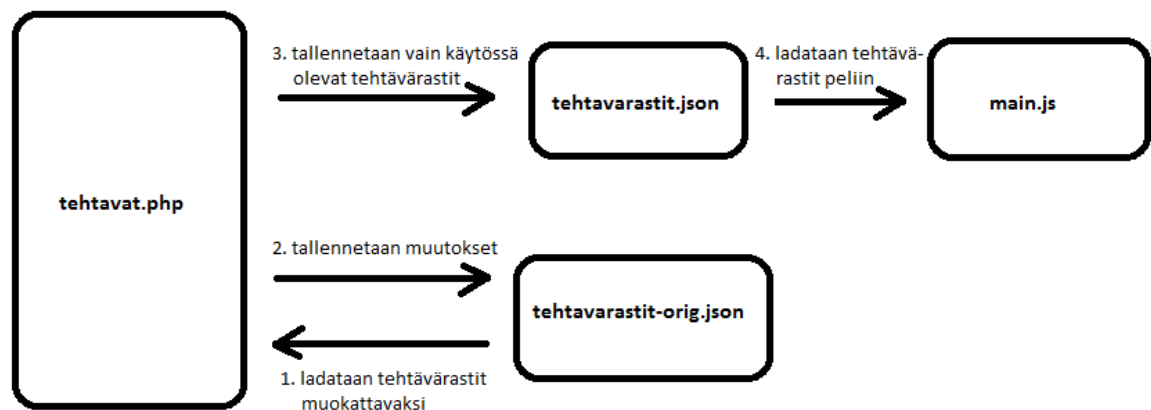
Tekstit	Kuvat	Tehtävät	Tehtävä-äänit	Muut äänit	Videot	Palkinnot
Tehtävärasti #1						
<input checked="" type="checkbox"/> Rasti on pelissä mukana						
Rastin nimi: <input type="text" value="Matkat"/>						
Siirtymäaika (sekuntia): <input type="text" value="8"/>						
Tehtävärasti #2						
<input checked="" type="checkbox"/> Rasti on pelissä mukana						
Rastin nimi: <input type="text" value="Ulko-ovella"/>						
Siirtymäaika (sekuntia): <input type="text" value="13"/>						
Tehtävärasti #3						
<input checked="" type="checkbox"/> Rasti on pelissä mukana						
Rastin nimi: <input type="text" value="Majat"/>						
Siirtymäaika (sekuntia): <input type="text" value="15"/>						
Tehtävärasti #4						
<input checked="" type="checkbox"/> Rasti on pelissä mukana						
Rastin nimi: <input type="text" value="Opetusryhmät - Pienet oppilaat"/>						
Siirtymäaika (sekuntia): <input type="text" value="11"/>						
Tehtävärasti #5						
<input checked="" type="checkbox"/> Rasti on pelissä mukana						
Rastin nimi: <input type="text" value="Kotitalous"/>						
Siirtymäaika (sekuntia): <input type="text" value="10"/>						

Kuvio 15. ”Tehtävät”-välilehti sisällönhallintatyökalussa

Tehtävärastien valintaa varten jouduttiin luomaan tehtavarastit.json-tiedoston rinnalle tehtavarastit-orig.json-tiedosto. Lisäksi JSON-tiedostoihin piti kullekin rastille lisätä ”kaytossa”-attribuutti. Tehtavarastit-orig.json-tiedostossa pidetään tallessa koko ajan kaikki olemassa olevat rastit. Siinä olevia tietoja kuitenkin muokataan sen mukaan,

miten käyttäjä on ne työkalussa asettanut. Tehtavarastit.json sisältää puolestaan ainoastaan rastit, jotka ovat pelissä käytössä.

PHP:ssä ladataan sivulle tehtavarastit-orig.json-tiedoston perusteella esitötetty lomake. Jos tehtävä on käytössä on sillä JSON:ssa "kaytossa"-attribuutin arvona 1 (muuten 0) ja tällöin myös lomakkeella "Rasti on pelissä mukana" -kohta on valmiiksi valittuna. Kun käyttäjä on tehnyt haluamansa muutokset ja lähettänyt lomakkeen, käydään se PHP:ssä rasti kerrallaan läpi. Kaikki muutokset tallennetaan tehtavarastit-orig.json-tiedostoon. Prosessin alussa tehtavarastit.json puolestaan tyhjennetään ja siihen lisätään ainoastaan käytössä olevat rastit. Näin toimimalla saadaan kaikki rastit pysymään tallessa tehtavarastit-orig.json:ssa ja kaikki käytössä olevat rastit voidaan ladata vanhaan tapaan tehtavarastit.json:sta pelin Javascriptissä ilman että Javascriptiä tarvitsi enää erikseen alkaa puukottamaan. Tätä prosessia havainnollistetaan kuviossa 16.



Kuvio 16. Tehtävärastien tallentaminen ja lataaminen

4.3.5 Tehtävä-äänien lataaminen

"Tehtävä-äänit"-välilehdellä käyttäjä voi ladata kaikki tehtäviin liittyvät äänet eli siirtymä-, kysymys- (6/rasti), vastaus- (6/rasti) ja vastausvaihtoehtoäänet (6*3/rasti). PHP:ssa luodaan sivulle jokaiselle käytössä olevalle rastille oma lomake äänien lataamiselle. Rastit ladataan tehtavarastit.json:n perusteella eli tällä välilehdellä näkyvät vaihtoehtoina vain käytössä olevat rastit.

Kun valitut äänitiedostot on lähetetty, tarkastetaan niiden oikeellisuus PHP:ssä. Jos tiedosto menee tarkastuksesta läpi, tallennetaan se oikeaan sijaintiin, joka saadaan tehtavarastit.jsonista. Kun se on tallennettu oikeaan paikkaan, toimii se pelissä automaattisesti. Kuvakaappaus ”Tehtävä-äänet”-välilehdestä näkyy kuviossa 17.

Tekstit	Kuvat	Tehtävät	Tehtävä-äänet	Muut äänet	Videot	Palkinnot
---------	-------	----------	----------------------	------------	--------	-----------

Tehtäväarasti #1: Matkat
Siirtymä-ääni (.mp3): Ei valittua tiedostoa

Tehtävä 1:
Kysymys (.mp3): Ei valittua tiedostoa
Vastaus kokonaisuudessaan (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (oikea vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa

Tehtävä 2:
Kysymys (.mp3): Ei valittua tiedostoa
Vastaus kokonaisuudessaan (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (oikea vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa

Tehtävä 3:
Kysymys (.mp3): Ei valittua tiedostoa
Vastaus kokonaisuudessaan (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (oikea vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa

Tehtävä 4:
Kysymys (.mp3): Ei valittua tiedostoa
Vastaus kokonaisuudessaan (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (oikea vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa

Tehtävä 5:
Kysymys (.mp3): Ei valittua tiedostoa
Vastaus kokonaisuudessaan (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (oikea vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa

Tehtävä 6:
Kysymys (.mp3): Ei valittua tiedostoa
Vastaus kokonaisuudessaan (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (oikea vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa
Vastausvaihtoehto (väärä vastaus) (.mp3): Ei valittua tiedostoa

Tehtäväarasti #2: Ulko-ovella
Siirtymä-ääni (.mp3): Ei valittua tiedostoa
Tehtävä 1:

Kuvio 17. ”Tehtävä-äänet”-välilehti sisällönhallintatyökalussa

4.3.6 Muiden äänien lataaminen

”Muut äänet”-välilehdellä käyttäjä voi ladata pelissä käytettävät oikeiden ja väärin vastausten jälkeen kuuluvat kannustusäänet sekä kolme ohjeistusääntä. Kannustusääniä on oikeille ja väärille vastauksille molemmille seitsemän ja ohjeistusääniä pelissä on kolme. Myös nämä äänitiedostot ladataan suoraan vanhojen tilalle, jolloin ne toimivat pelissä suoraan ilman muita muutoksia. Kuvakaappaus ”Muut äänet”-välilehdestä näkyy kuviossa 18.

Kannustukset oikeasta vastauksesta:

Kannustus 1:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Kannustus 2:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Kannustus 3:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Kannustus 4:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Kannustus 5:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Kannustus 6:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Kannustus 7:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa

Ilmoitukset väärästä vastauksesta:

Ilmoitus 1:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ilmoitus 2:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ilmoitus 3:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ilmoitus 4:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ilmoitus 5:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ilmoitus 6:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ilmoitus 7:	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa

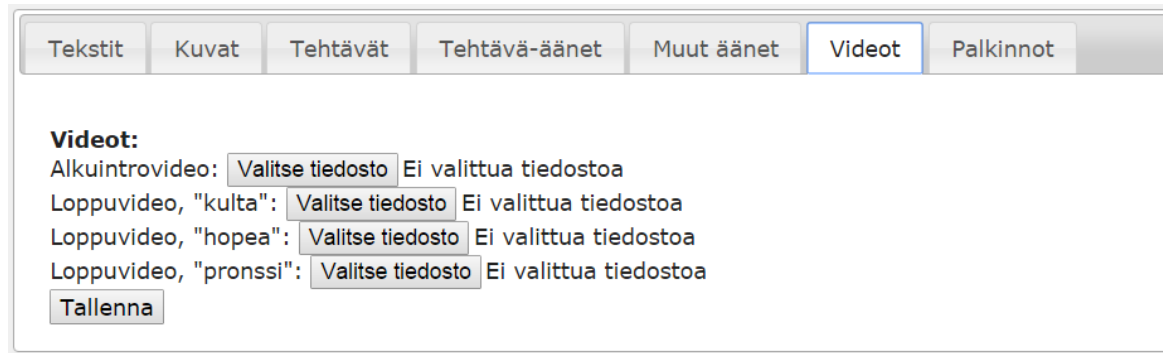
Pelin ohjeistusänet:

Ohjeistus 1 (heti alkuvideon jälkeen yleisnäkymässä kuuluva):	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ohjeistus 2 (ensimmäiseen tehtävään mennessä, ennen tehtävän kuulumista):	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa
Ohjeistus 3 (ensimmäisen tehtävän kuulemisen jälkeen):	<input type="button" value="Valitse tiedosto"/>	Ei valittua tiedostoa

Kuvio 18. ”Muut äänet”-välilehti sisällönhallintatyökalussa

4.3.7 Videoiden lataaminen

Videoiden lataamisessa on käytössä sama periaate kuin edellisissäkin kohdissa. Käyttäjä voi ladata peliin alkuintrovideon sekä kierroksen lopussa näytettävät kulta-, hopea- ja pronssivideot. Kuvakaappaus tästä välilehdestä näkyy kuviossa 19.



Tekstit Kuvat Tehtävät Tehtävä-äänet Muut äänet **Videot** Palkinnot

Videot:

Alkuintrovideo: Valitse tiedosto Ei valittua tiedostoa

Loppuvideo, "kulta": Valitse tiedosto Ei valittua tiedostoa

Loppuvideo, "hopea": Valitse tiedosto Ei valittua tiedostoa

Loppuvideo, "pronssi": Valitse tiedosto Ei valittua tiedostoa

Tallenna

Kuvio 19. "Videot"-välilehti sisällönhallintatyökalussa

4.3.8 Palkintojen hintojen muuttaminen

Etenkin koska peliin voi valita eri määriä tehtäviä, on oleellista, että myös pisteillä ostettavien palkintojen hinnat pystyy suhteuttamaan jaossa olevien pisteiden määrään. Palkintojen hinnat pystyy määrittelemään "Palkinnot"-välilehdellä. Siellä oleva lomake on esitäytetty prizes.json-tiedostosta saaduilla tiedoilla. Kun käyttäjä tallentaa tekemänsä hintamuutokset, ne tallentuvat samaiseen JSON-tiedostoon, josta ne sitten ladataan Javascriptissä peliin. Kuvakaappaus välilehdestä näkyy kuviossa 20.

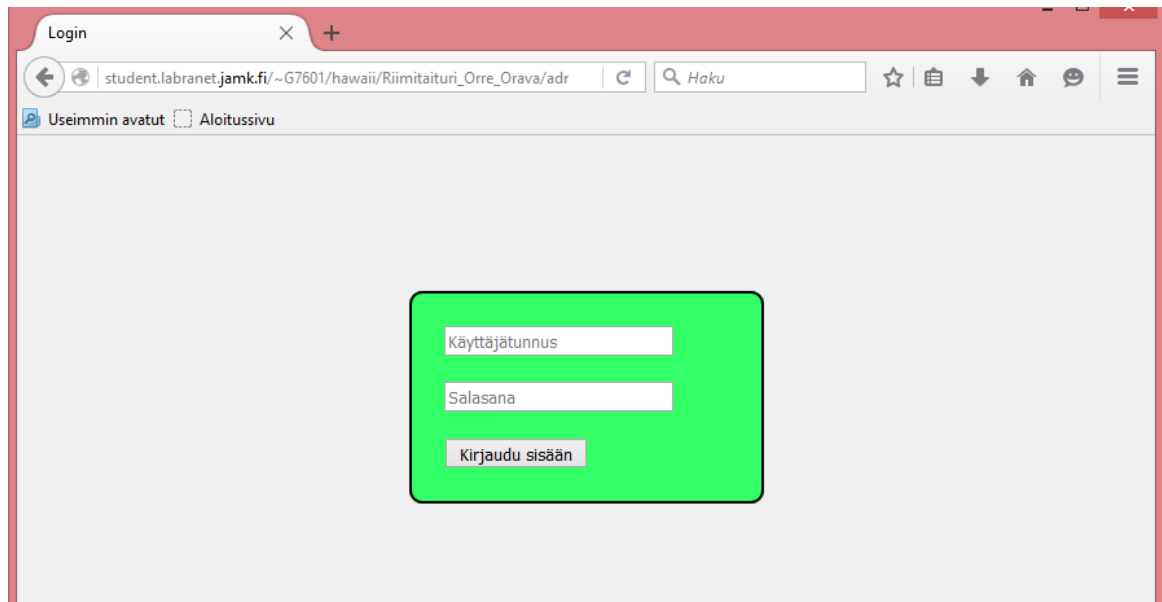
Tekstit	Kuvat	Tehtävät	Tehtävä-äänet	Muut äänet	Videot	Palkinnot
Palkinto: shortsit						
Palkinnon hinta: <input type="text" value="2"/>						
Palkinto: hattu						
Palkinnon hinta: <input type="text" value="6"/>						
Palkinto: aurinkolasit						
Palkinnon hinta: <input type="text" value="10"/>						
Palkinto: surffilauta						
Palkinnon hinta: <input type="text" value="8"/>						
Palkinto: rantapallo						
Palkinnon hinta: <input type="text" value="4"/>						
Palkinto: kukat						
Palkinnon hinta: <input type="text" value="3"/>						
Palkinto: riippumatto						
Palkinnon hinta: <input type="text" value="8"/>						
Palkinto: laiva						
Palkinnon hinta: <input type="text" value="11"/>						
Palkinto: talo						
Palkinnon hinta: <input type="text" value="12"/>						
Palkinto: aurinko						
Palkinnon hinta: <input type="text" value="9"/>						
<input type="button" value="Tallenna"/>						

Kuvio 20. ”Palkinnot”-välilehti sisällönhallintatyökalussa

4.3.9 Sisäänkirjautuminen

Sisällönhallintatyökalu ei luonnollisestikaan saa olla kaikkien käytettävissä, jottei sen avulla pääse tekemään ilkivaltaa ja sotkemaan pelin sisältöä. Varmin keino tämän estämiseksi on tietysti poistaa koko työkalu siinä vaiheessa, kun peli on saatu valmiiksi ja julkaistaan. Toisaalta jos työkalu halutaan pitää valmiin pelin muokkamista varten, kannattaa sille pääsy rajoittaa tietyille käyttäjille palvelinpuolelta.

Työkalulle on toteutettu myös yksinkertainen käyttäjätunnuksella ja salasalla sisäänkirjautuminen. Tunnus ja salasana on kovakoodattu suoraan PHP-koodiin. Kun käyttäjä kirjautuu tällä yhdistelmällä sisään, saa hän käyttöönsä session, jolla työkalu on käytettävissä. Kuviossa 21. on esitetty kuvakaappaus sisäänkirjautumisnäkyvästä.



Kuvio 21. Sisällönhallintatyökalun sisäänkirjautumisnäkymä

5 TULOKSET JA LOPPUPOHDINTA

Opinnäytetyön lopputuloksena saatiin yleisen tason raportti verkkosovellusten ja -sivujen optimointimenetelmistä. Lisäksi Riimitaituri Orre Orava saatiin korjattua julkaisukelpoiseen kuntoon ja peliin saatiin myös lisättyä muutamia lisäominaisuuksia.

Verkkosovellusoptimointikeinoja saatiin selvitettyä hyvä määrä ja niitä kokeiltiin myös käytännössä. Ajanpuutteen vuoksi optimointiin ei kuitenkaan pystytty menemään niin syvälle kuin oli alunperin tarkoitus. Esimerkiksi JavaScript-optimoinnin käsittely jäi turhan vähälle huomiolle.

Opinnäytetyön ehkä paras anti oli kuitenkin käytännön toteutuksissa Riimitaituri Orre Orava –pelin parissa. Kaikki kriittisimmät virheet saatiin korjattua ja näin pelistä tuli julkaisukelpoinen. Peliin lisätty palkintokauppa toi mukavaa lisäsisältöä ja tavoiteltavaa pelaajalle. Erityisen hyödyllinen pelin jatkohyödyntämistä varten on opinnäytetyön aikana toteutettu sisällönhallintatyökalu. Tämä mahdollistaa uusien pelien toteuttamisen vanhan pelin runkoa ja mekaniikkaa hyödyntäen ilman, että käyttäjän tarvitsee mennä muokkaamaan koodia.

Valitettavasti käytännön osuudessa peliä ei onnistuttu keventämään niin paljon, että se olisi toiminut iPad2:lla tai muilla hieman vanhemmilla laitteilla. Peliä saatiin kuitenkin jonkin verran optimoitua ja se toimii nyt sujuvammin kuin aikaisemmin.

Kokonaisuudessa työhön voi olla kohtuullisen tyytyväinen. Työ saatiin valmiiksi suunnitellussa aikataulussa ja sille asetetut vähimmäistavoitteet saavutettiin, vaikka ajankäytön kanssa oli huomattavia haasteita muiden kiireiden takia.

LÄHTEET

Best Practices for Speeding Up Your Web Site. N.d. Artikkelin Yahoo! Developer Networkissa. Viitattu 13.12.2015. <https://developer.yahoo.com/performance/rules.html>

Grigorik, I. N.d.a HTTP Caching. Viitattu 10.2.2016.
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/http-caching>

Grigorik, I. N.d.b Image optimization. Viitattu 13.12.2015.
<https://developers.google.com/web/fundamentals/performance/optimizing-content-efficiency/image-optimization>

Guasch, C. 2012. What is Web Performance Optimization? (WPO). Viitattu 13.12.2015.
<http://senkailabs.com/optimizers/what-is-web-performance-optimization/>

Rumpel, C. 2013. Front-end performance part 01: Assets loading. Viitattu 8.2.2016.
<http://christoph-rumpel.com/2013/04/front-end-performance-part-01-assets-loading/>

Sexton, P. 2015. Enable gzip compression. Viitattu 16.1.2016.
<https://varvy.com/pagespeed/enable-compression.html>

Smits, W. 2014. How to minimize HTTP requests. Viitattu 13.12.2015.
<http://www.wesleysmits.com/how-to-minimize-http-requests/>

Valteri-koulu Onerva. Koulun esittelysivu Onervan kotisivuilla. Viitattu 17.12.2015.
<http://www.onerva.fi/valteri-koulu-onerva/>